Approximate Greatest Descent Method and Quasi-Newton Matrices in Optimization

B. S. Goh\textsuperscript{1}, Zheng Peng\textsuperscript{2}, C.S. Lee\textsuperscript{3}, Junfeng Yang\textsuperscript{4}, Min Kong\textsuperscript{4}

\textsuperscript{1}Institute of Mathematical Sciences, University of Malaya, 50603 Kuala Lumpur, Malaysia.
\textsuperscript{2}College of Mathematics and Computer Science, Fuzhou University, 350108, Fuzhou, P. R. China.
\textsuperscript{3}Former Associate Professor, Mathematics and Mathematics Education, NIE, Nanyang Technological University, Singapore.
\textsuperscript{4}Mathematics Department, Nanjing University, Nanjing, Jiangsu 210093, China.

Abstract. The long-term optimal trajectory to compute a minimum point consists of a sequence of greatest descent steps followed by the Newton step in the last iteration. The greatest descent direction can be approximated by a Levenberg-Marquardt like formula. There is a simple way to prescribe the relative steplengths so that the approximate greatest descent (AGD) direction merges into the Newton direction near a minimum point. This enables fast local convergence of the AGD method near a minimum point. Here we examine the use of the $B_k$ matrices, defined by a quasi-Newton update formula, as a way to model and approximate the Hessian matrix of a nonlinear function. These $B_k$ matrices are used in the AGD iteration rather than the Newton iteration. Used in this manner numerical errors in the $B_k$ matrices can be tolerated when the point is at a large distance from the minimum point. Furthermore, $B_k$ is not required to be positive definite or nonsingular. Instead we require a weaker condition, namely the function is monotonic decreasing. This can always be achieved by using a small steplength. Computational errors can make the $B_k$ matrix singular and not positive definite. From numerical experiments the main advantage of using the $B_k$ matrix rather than the Hessian matrix in the AGD method is that it is faster when the number of variables is large.

Keywords. Unconstrained optimization; greatest descent direction; Newton method; quasi-Newton method; convergence.

AMS (MOS) subject classification: 90C30, 97N60, 65K05.

1 Introduction

Goh (2009) has shown that the structure of a long term optimal trajectory to compute the minimum point of an unconstrained function consists of a sequence of greatest descent steps and the Newton method as the last step. The

\textsuperscript{1}B.S. Goh, Corresponding author, Professor, Institute of Mathematical Sciences, University of Malaya, 50603 Kuala Lumpur, Malaysia. E-mail: goh2optimum@gmail.com
greatest descent step can be implemented approximately with a Levenberg-Marquardt like formula. An important success factor in this new approach is the way the relative steplengths are chosen so that they increase in an automatic manner as the current point approaches the minimum point. These relative steplengths become very large near the minimum point. When the relative steplength tends to infinity, the approximate greatest descent (AGD) direction merges into the Newton direction. The relative steplength is set equal to a positive constant divided by the Euclidean norm of the gradient vector of the objective function.

With this new approach it follows that the Newton iteration is only important near a minimum point to provide rapid convergence in the final iterations. At a significant finite distance from a minimum point, the AGD iterations play the central role.

Here we propose that the Hessian matrix in the Levenberg-Marquardt like formula be replaced by a $B_k$ matrix which is defined by one of the update formulae known as SRI, DFP, BFGS and others, see Nocedal and Wright (1999). At the current point $x_k$ the BFGS update formula is

$$B_{k+1} = B_k - \frac{(B ks_k)(s_k^T B_k)}{s_k^T B k s_k} + \frac{y_k y_k^T}{y_k^T s_k},$$  

where $B_1 = I$, the identity matrix, $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$ and where $g_k = \nabla f(x_k)$ is the gradient vector.

These $B_k$ matrices tend to stabilize and become constant matrices when the current point is close to a minimum point. Furthermore, near the minimum point they approximate the Hessian matrix. However, at a finite distance from the minimum point the values of the components of these $B_k$ matrices can change rapidly. We shall show graphically that in general at a significant finite distance from a minimum point there is little relationship between the Hessian matrix of a function and these $B_k$ matrices. Thus at such points it is more correct to say that these $B_k$ matrices are models of the Hessian matrix rather than to say that they approximate the Hessian matrix. In particular at the first iteration, $B_1 = I$ and the Hessian matrix could then be a matrix which cannot be approximated in any way by an identity matrix. From numerical examples of nonquadratic functions we find that in general these $B_k$ matrices approximate the Hessian matrix only when the current point is close to the solution.

By using the AGD iteration rather than the Newton iteration with these $B_k$ matrices, we can automatically build into the algorithm some tolerance to the errors in computing these $B_k$ matrices at points, which are at a finite distance from the minimum point. At such a point, the identity matrix in the Levenberg-Marquardt like formula can dominate the $B_k$ matrix. Near the minimum point these $B_k$ matrices stabilizes and become constant matrices which converge to the Hessian matrix of the function at the minimum point. Thus near the minimum point, they are less prone to numerical errors. There
these $B_k$ matrices can then dominate the identity matrix in the Levenberg-Marquardt like formula without causing serious difficulties. Furthermore, we shall not require the $B_k$ matrix to be positive definite at a point which is at a finite distance from the minimum point. We shall only require the weaker condition that the function decreases monotonically. Finally we can define and prove convergence of the AGD method with the $B_k$ matrix for a general nonlinear objective function. According to Nocedal and Wright (1999) in page 211 such a convergence analysis has not been achieved with the standard quasi-Newton method.

2 Approximate greatest descent method with $B_k$ matrices

We want to minimize the unconstrained function $f(x)$ where $x \in \mathbb{R}^n$ and it has continuous second order derivatives. Assume that it has a unique minimum at $x^*$ in a closed bounded region $\Omega \in \{x | f(x) \leq K\}$, where $K$ is a constant.

The iterative equation is

$$x_{k+1} = x_k + s_k = x_k + \alpha_k p_k, \text{ where } k = 1, 2, \ldots$$ (2)

Here $\alpha_k$ is the relative steplength and $p_k$ is the search direction. We should call $\alpha_k$ the relative steplength because the norm of $p_k$ may not be equal to one. The actual step length is $\beta_k = \alpha_k \|p_k\|$. However, when convenient, we shall follow the common practice and also call $\alpha_k$ the steplength. Let the gradient vector be denoted by $g_k = \nabla f(x_k)$ and the Hessian matrix by $G_k = \nabla^2 f(x_k)$.

In an approximate greatest descent algorithm, we have

$$x_{k+1} = x_k + s_k = x_k - \alpha_k [I + \alpha_k G_k]^{-1} g_k.$$ (3)

An important success factor in our proposed method is that the relative steplength $\alpha_k$ is defined by

$$\alpha_k = \frac{\Delta}{\|g_k\|}.$$ (4)

Here $\Delta$ is a positive parameter which is an approximation of the radius of a local spherical search region as in a trust region method. Usually we set $\Delta = \|g_1\|$ or $\Delta = 1$. The steplength formula in (2.3) or equivalent has been used by Kelley (1999), Grantham (2003) and Fan and Yuan (2005). It adjusts itself automatically in the right way as the current point approaches the minimum point. At the minimum point, the gradient vector is equal to the zero vector. In this manner, the relative step length $\alpha_k$ tends to infinity as the point approaches the minimum point. When this happens equation (2.2) of the proposed method converges to the Newton method and fast convergence is achieved.
To simplify the iterative equation (2.2) we use

$$\mu_k = \frac{1}{\alpha_k} = \frac{||g_k||}{\Delta}$$

and the iterative equation becomes

$$x_{k+1} = x_k + s_k = x_k - [\mu_k I + G_k]^{-1} g_k.$$ (6)

We now study the approximate greatest descent method with the Hessian matrix $G_k$ replaced by a $B_k$ matrix from one of the formulae in a quasi-Newton method, see Nocedal and Wright (1999). To illustrate this approach we shall focus on the use the BFGS formula. Let $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$. We have

$$B_{k+1} = B_k - \frac{(B_k s_k)(s_k^T B_k)}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k},$$

where $B_1 = I$, the identity matrix.

Traditionally, this $B_k$ is used with the Newton direction,

$$s_k = -\alpha_k B_k^{-1} g_k,$$ (8)

where $\alpha_k$ is the relative steplength and (2.7) is used in the iterative equation (2.1). We propose to use an AGD step $s_k$ given by

$$[\mu_k I + B_k]s_k = -g_k.$$ (9)

Thus, we have

$$x_{k+1} = x_k + s_k = x_k - [\mu_k I + B_k]^{-1} g_k, \text{ where } k = 1, 2, \ldots$$ (10)

Here we note that in (2.8) the search direction vector $s_k$ is a function of the relative steplength $\alpha_k$ or $\mu_k$. Thus, the relative steplength is first chosen and then the search direction is computed. We then check that the value of the function decreases. If not, the relative steplength is reduced repeatedly. We note that as the relative steplength $\alpha_k$ is reduced or $\mu_k$ is increased, the search direction merges into the steepest descent direction at the current point.

We expect the matrix $B_k$ to be stable and that it is not prone to errors near the minimum point. In fact, $B_k$ should converge to the Hessian matrix at the minimum point. However, at a significant finite distance from the minimum point, we can in general expect that the values of the elements of the matrix $B_k$ to change rapidly. We demonstrate this behavior by a study of the values of $\det(B_k)$ along a trajectory with the Rosenbrock function. We compare the values of the determinants of $B_k$ and $G_k$ and show that they can be different qualitatively and quantitatively. Thus at such points it is
more correct to say that the $B_k$ matrix models the $G_k$ matrix rather than to say that the $B_k$ matrix approximates the $G_k$ matrix.

There are several advantages of the AGD direction defined by (2.8) over the Newton method direction (2.7). Firstly, the long-term optimal trajectory is made of a sequence of AGD iterations followed by the Newton step only in the last iteration. Secondly, at a point, which is at a finite distance from the minimum point, the values of the components of the matrix $B_k$ can change rapidly and be more prone to errors. With small values in the relative steplengths $\{\alpha_k\}$, the term $\mu_k I$ in (2.9) can be large and damps out such errors. Thirdly, in this new approach the matrix $B_k$ is not required to be positive definite. It is only required to satisfy the weaker monotonic function decrease condition,

$$\Delta f(x_k) = f(x_k + s_k) - f(x_k) < 0$$  \hspace{1cm} (11)

at points away from the minimum point.

We use these ideas to construct an algorithm. We need some housekeeping conditions to ensure that the function decreases monotonically.

**Algorithm 2.1.** Approximate greatest descent method with a $B_k$ matrix.

1. Choose small number $\varepsilon_1$ to control termination of the iterations and $\varepsilon_2$ to test singularity of a matrix by the value of its determinant. Choose an initial point $x_1$. Choose a contraction factor $d$ where $(0 < d < 1)$, say $d = 0.5$.

2. Initial iteration: Choose $\Delta = \|g_1\|$ or $\Delta = 1$, set $\alpha_1 = \frac{\Delta}{\|g_1\|}$ and $p_1 = -g_1$. Test

$$\Delta f(x_k) = f(x_k + \alpha_1 p_1) - f(x_k) < 0.$$  \hspace{1cm} (12)

If (12) is not satisfied $\alpha_1$ is replaced repeatedly by $d\alpha_1$ until (12) is satisfied.

3. Set $B_1 = I$, the identity matrix. Define $B_k$ by the BFGS formula (2.6).

4. Let $\mu_k = \frac{\|g_k\|}{\Delta x}$, if the $\det[\mu_k I + B_k]$ is less than $\varepsilon_2$ the parameter $\mu_k$ is replaced repeatedly by $\mu_k/d$ until this determinant is greater than $\varepsilon_2$.

5. Let the direction vector $s_k = -[\mu_k I + B_k]^{-1}g_k$. Test that

$$\Delta f(x_k) = f(x_k + s_k) - f(x_k) < 0.$$  \hspace{1cm} (13)

If (13) is not satisfied $\mu_k$ is replaced repeatedly by $\mu_k/d$ until (13) is satisfied. For each value of $\mu_k$ the direction vector $s_k$ needs to be re-calculated.
6. Iteration is terminated if \( \|g_k\| < \varepsilon_1 \).

**Notes:** In the literature on Lyapunov function, see, LaSalle (1979), Kalman and Bertram (1960) Vincent and Grantham (1997) and Goh (2010), we say that a function \( V(x) \) is positive definite in a region if \( V(x) > 0 \) for \( x \neq x^* \) and \( V(x^*) = 0 \). A function \( V(x) \) is negative definite if \( -V(x) \) is positive definite. A function \( V(x) \) is radially unbounded if \( V(x) \rightarrow \infty \) as \( \|x - x^*\| \rightarrow \infty \).

**Convergence.** We now describe a set of conditions for convergence. Assume:

(i) The function \( f(x) \in C^2 \) is bounded below and that it has a unique stationary point \( x^* \) in the region \( \Omega = \{x | f(x) - f(x^*) \leq L\} \), where \( L \) is an arbitrary large positive number.

(ii) The function \( f(x) \) is radially unbounded, a condition that is needed for \( V(x) = f(x) - f(x^*) \) to be a Lyapunov function globally. The candidate Lyapunov function \( V(x) \) inherits these properties from \( f(x) \).

(iii) There exists a positive constant such that

\[
\Delta V(x_k) = \Delta f(x_k) = f(x_k + s_k) - f(x_k) < -c_3 g_k^T g_k \tag{14}
\]

for all \( x_k \neq x^* \) is satisfied. The constant \( c_3 \) can be a very small positive number.

When conditions (i) to (iii) are satisfied we get convergence by the Lyapunov Function Theorem, see LaSalle (1979) and Goh (2010).

By construction, steps (2) and (5) in Algorithm 2.1 ensure that the difference function \( \Delta V(x_k) = \Delta f(x_k) < 0 \) if \( x_k \neq x^* \). Furthermore, when the steplength is sufficiently small the search direction is approximated by the steepest descent direction. A strengthened Armijo’s monotonic decrease condition (2.13) is imposed to strengthen step 5 so that the condition \( \Delta V(x_k) = \Delta f(x_k) < 0 \) is independent of \( k \).

Condition (2.13) has been used by Gilbert and Nocedal (1992), Shi and Shen (2004) to study global convergence of algorithms in optimization. They called it the sufficient descent condition. It is a very useful condition as it enables us to invoke and apply the Lyapunov Function Theorem for convergence of Algorithm 2.1.

Note that \( s_k = -(\mu_k I + B_k)^{-1} g_k \) is an explicit function of the time variable \( k \) because of the update BFGS formula (2.6). Thus (2.9) is a time varying dynamical system. We need (2.13) to apply Lyapunov Function Theorem to a time varying system, see LaSalle (1979). Condition (2.13) implies \( \Delta V(x_k) = \Delta f(x_k) < -c_3 g_k^T g_k \) is satisfied independently of the iteration index (the time variable as a dynamical system) \( k \). It is a property of only the point \( x_k \).

Thus \( \Delta V(x_k) \) is negative definite in \( \Omega = \{x | V(x) \leq L\} \). As \( L \) is an arbitrary and large positive number, we have global convergence for Algorithm
2.1 under assumptions (i) to (iii). This is a consequence of Lyapunov Function Theorem for a time varying discrete-time system (iterative equations), see LaSalle (1979).

Notes: In Algorithm 2.1, the relative steplength \( \alpha_k \to \infty \) (or \( \mu_k \to 0 \)) as the point approaches the minimum point. In the limit, it converges to the Newton method and fast convergence is achieved.

We expect that steps 2 and 5 would ensure that condition (2.13) is satisfied in most cases. This is because the steepest descent direction would satisfy (2.13). Here, when the steplength is small, the search direction approximates the steepest descent direction. Steps 2 and 5 ensure that the decreases monotonically and does so independent of the iteration number \( k \).

Section 8.4 of the well known book by Nocedal and Wright (1999) on the convergence of quasi-Newton methods states that “we will not be able to establish truly global convergence results for general nonlinear objective functions”. By using the \( B_k \) matrix in an AGD iteration rather than a Newton iteration we have found a way to establish global convergence of the new method for a general nonlinear objective function which is comparable in generality to a steepest descent method. This is because the AGD iteration merges into the steepest descent iteration when the parameter \( \mu_k \to \infty \).

It is interesting to note that for convergence of algorithm 2.1 the matrix \( B_k \) need not be positive definite or nonsingular. We only require the weaker condition that the function is monotonic decreasing. This can always be achieved by taking a sufficiently small steplength. Then the AGD iteration merges into the steepest descent iteration. However like a steepest descent method we may have theoretical convergence but it can fail in practice because of numerical errors.

As expected, the matrix \( B_k \) generated by Algorithm 2.1 is different from the matrix \( B_k \), which is generated when the Newton direction is used. We shall illustrate this by comparing the values of the determinants of these \( B_k \) matrices when the AGD direction and the Newton direction are used. This will be shown in the numerical example below.
Example. 2.1. Consider the Rosenbrock function \( f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \). The initial point is \((-1, 1.005)\). We have chosen this point instead of the standard initial point because the Hessian matrix is singular at this point. With \( \varepsilon_1 = 10^{-7} \), it converges in 39 iterations. The function is radially unbounded as \( f(x) \to \infty \) as \( \|x - x^*\| \to \infty \). The problem has a unique stationary point. We have global convergence as (2.13) can be satisfied with exact computations. We have fast convergence as the method merges into the Newton method near the equilibrium point.

It is interesting to compare the values of the determinant of \( B_k \) and the determinant of the Hessian matrix \( G_k \) in Algorithm 2.1. Away from a finite distance from the minimum point, these two determinants differ in their qualitative and quantitative behaviors, see Fig. 1. Thus at such points we should say that the matrix \( B_k \) is a model of \( G_k \). The matrix \( B_k \) is not an approximation of \( G_k \) at such points. Near the minimum point, the values of these two determinants merge.
Again Fig.2 shows that the determinants of matrix $B_k$ and the Hessian matrix $G_k$ generated by the quasi-Newton iteration are different. The initial point (-1,1.005) is used. At a significant finite distance from the solution these two determinants are different qualitatively and quantitatively. Thus again at such points we should say that the matrix $B_k$ is a model of $G_k$. The matrix $B_k$ is not an approximation of $G_k$ at such points. Near the minimum point, the values of these two determinants merge.

Furthermore from Fig.1 and 2 the matrix $B_k$ which is generated by the AGD iteration in Algorithm 2.1 is different from the matrix $B_k$ which is generated by a quasi-Newton method using the Newton iteration with the $B_k$ matrix.

We examined the trajectories of the quasi-Newton method with a backtracking line search from the initial point (-20,30). For the contraction factor $d = 0.1$ the matrix $B_k$ becomes singular in practice even though theoretically it is supposed to remain positive definite. This may be caused by numerical errors in computation. On the other hand, Algorithm 2.1 with the AGD iteration and the $B_k$, converges for a wide range of values of the contraction factor. This shows that the AGD iteration with the $B_k$ may be more robust than the standard quasi-Newton method.

**Example 2.2.** We tested the new algorithm on 72 problems from Andrei (2008). For problems with less than 50 variables the AGD method with the Hessian matrix is faster than the AGD method with the $B_k$ matrix from the BFGS update formula. For test problems with more than 100 variables the AGD method with $B_k$ matrix is faster. This is because the time needed to compute the $B_k$ is generally less than the time to compute the Hessian matrix.
The CPU time for the AGD method with the $B_k$ matrix is comparable with that for the quasi-Newton method. Table 1 shows the results for a subset of ten problems.

<table>
<thead>
<tr>
<th>Function name</th>
<th>dim.</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$n$</td>
<td>AGD-$G_k$</td>
</tr>
<tr>
<td>Extended Trigonometric</td>
<td>100</td>
<td>4.01E2</td>
</tr>
<tr>
<td>Extended White &amp; Holst</td>
<td>100</td>
<td>1.46E2</td>
</tr>
<tr>
<td>Extended Beale</td>
<td>100</td>
<td>4.50E1</td>
</tr>
<tr>
<td>Extended QP-1</td>
<td>100</td>
<td>6.51E1</td>
</tr>
<tr>
<td>Extended QP-2</td>
<td>100</td>
<td>8.18E1</td>
</tr>
<tr>
<td>DIXMAANA</td>
<td>120</td>
<td>6.03E1</td>
</tr>
<tr>
<td>DIXMAANB</td>
<td>120</td>
<td>8.39E1</td>
</tr>
<tr>
<td>DIXMAANC</td>
<td>120</td>
<td>8.03E1</td>
</tr>
<tr>
<td>ARWHEAD</td>
<td>100</td>
<td>3.36E1</td>
</tr>
<tr>
<td>DIXMAANE</td>
<td>120</td>
<td>1.37E3</td>
</tr>
</tbody>
</table>

Table 1: Test results for a subset of ten problems.

**Remark:** In all numerical test, termination condition is $\|g_k\| < 10^{-7}$. All test problems use the standard initial point of each problem. The number of variables is $n$.

### 3 Conclusions

From numerical experiments with the Rosenbrock function Figs.1 and 2 show that at points at a significant finite distance from the solution the $B_k$ matrix in the AGD method or the quasi-Newton method does not approximate the Hessian matrix when the objective function is nonlinear. At such points it is better to say that the $B_k$ matrix is a model of the Hessian matrix. The $B_k$ is generated by the BFGS update formula.

From numerical experiments with 72 test problems we find that the AGD method with the Hessian matrix is faster than the AGD method with the $B_k$ matrix when the number of variables is small. For problems with more than one hundred variables the AGD method with the $B_k$ matrix is faster. This is because it is increasingly more costly time-wise to compute the Hessian matrix. We find that there is no difference in performance between the AGD method with the $B_k$ matrix and Newton method with the $B_k$ matrix. The main advantage is that we can establish and prove convergence for a general nonlinear objective function. Furthermore the region of convergence is comparable with that for a steepest descent method. This is achieved because the AGD direction merges into the steepest descent direction when the steplength is small.
Using the Rosenbrock function we show that the quasi-Newton method with the $B_k$ matrix from the BFGS update formula can fail from the initial value (-20,30). Theoretically the matrix $B_k$ should remain positive definite. It may be that numerical errors in computation have caused it to become singular. On the other hand the approximate greatest decent iteration with the $B_k$ matrix from the BFGS update formula converges without any difficulty from the initial value (-20, 30). The result is expected because the AGD iteration does not require the $B_k$ to be positive definite or nonsingular. It requires the weaker condition that the function is monotonic decreasing. This can always be satisfied as it merges into the steepest descent iteration with a small steplength.

4 Acknowledgements

We thank Professor Bing Sheng He and Mark Wu for their help and support in this research. The work of Junfeng Yang was supported by the Natural Science Foundation of China NSFC-11001123.

5 References


Received November 2009; revised January 2010; March 2010; June 2010.

email: journal@monotone.uwaterloo.ca
http://monotone.uwaterloo.ca/~journal/